



# 07. Estructuras de datos

Programación - 1º DAM

Luis del Moral Martínez

versión 20.10

Bajo licencia CC BY-NC-SA 4.0



# Contenidos del tema

1. Estructuras de datos
2. Arrays de estructuras

# 1. Estructuras de datos

## Concepto de estructura de datos

- Un array, o vector, es una **sucesión** de objetos del mismo tipo
- Si necesitamos almacenar tipos de datos diferentes al mismo tiempo, usaremos una **estructura**
- Una **estructura** es una colección de uno o más elementos o **miembros**
- Cada elemento o miembro puede tener un **tipo de dato** diferente

# 1. Estructuras de datos

## Declaración de una estructura

- Una estructura es un **tipo de dato** definido por el usuario

```
struct nombre_estructura
{
    tipo_dato miembro1;
    tipo_dato miembro2;
    tipo_dato miembro3;
    ...
};
```

```
struct cd_rom
{
    char titulo[30];
    char artista[25];
    int num_canciones;
    float precio
};
```

# 1. Estructuras de datos

## Definir una variable de tipo estructura

- Podemos crear una variable del tipo de dato estructura de dos formas:

### Opción 1

```
struct cd_rom  
{  
    char titulo[30];  
    char artista[25];  
    int num_canciones;  
    float precio  
} cd1, cd2, cd3;
```

### Opción 2

```
struct cd_rom  
{  
    char titulo[30];  
    char artista[25];  
    int num_canciones;  
    float precio  
};  
  
cd_rom cd1, cd2, cd3;
```

# 1. Estructuras de datos

## Inicialización de una estructura

- Existe la posibilidad de inicializar una estructura en el momento que se declara

```
struct Date  
{  
    int month;  
    int day;  
    int year;  
};
```

```
Date birthdate = {1, 1, 2010};
```

# 1. Estructuras de datos

## Acceso a una estructura

- Podemos acceder a una estructura mediante dos operadores:
  - **Operador punto**: es la opción por defecto
  - **Operador puntero (->)**: se estudiará más adelante, en el capítulo de punteros
- Abre el fichero de ejemplo **07\_01\_estructuras.cpp**

## 2. Arrays de estructuras

### Creando un vector de estructura

- Es posible crear un vector usando una **estructura como tipo de dato**
- La declaración es similar a la de cualquier array: **tipo\_dato nombre[elementos];**
- El acceso y la definición es igual, pero teniendo en cuenta el número de elementos del array
- Abre el fichero de ejemplo **07\_02\_array\_estructuras.cpp**



# Créditos de las imágenes y figuras

## Cliparts e iconos

- **Obtenidos mediante la herramienta web [IconFinder](#)** (según sus disposiciones):
  - Diapositiva 1
  - Según la plataforma IconFinder, dicho material puede usarse libremente (free comercial use)
  - A fecha de edición de este material, todos los cliparts son free for comercial use (sin restricciones)

## Resto de diagramas y gráficas

- Se han desarrollado en PowerPoint y se han incrustado en esta presentación
- Todos estos materiales se han desarrollado por el autor
  - Si se ha empleado algún icono externo, este se rige según lo expresado anteriormente